



**Forum: WD 11**

**Topic: d&#039;éviter le popup explorateur fichier lors d&#039;insertion disques**

**Subject: d'eviter le popup explorateur fichier lors d'insertion disques**

PostÃ© par: Chapeau\_m

Contribution le : 8/11/2007 14:01:05

Bonjour,

apriori il n'y a pas moyen d'éviter le popup explorateur fichier lors d'insertion disques (usb ou autres).

Surtout que je ne veux pas jouer avec des changements de registre windows.

Par contre j'ai décelé dans le programme Zexplorateur (zapsolution.com - source libre) écrit en Windev, qu'il y a quand même une solution :

Je n'ai pas besoin de l'explorateur.

Pourrait-on m'aider à filtrer juste le code qui sert à éviter le popup ??

Dans mon projet je travaille avec fDétekteStockageAmovible car cela me convient parfaitement.

BAV

WM\_QUERYCANCELAUTOPLAY est entier = API("USER32", "RegisterWindowMessageA", "QueryCancelAutoplay")

// Et on bloque la popup Windows d&#039;insertion automatique (CD-ROM/DVD)  
Événement("DeviceNotification", FenEnExécution(), WM\_QUERYCANCELAUTOPLAY)

```
// ***** // La fonction WinDev  
fDétekteStockageAmovible étant trop limitée // On gère nous-même l&#039;insertion ou la  
suppression des devices amovibles // *****  
FONCTION DeviceNotification(wMsg, wParam, lParam) DEV_BROADCAST_HDR est structure  
dbch_size est un entier sans signe sur 4 octets dbch_devicetype est un entier sans signe sur 4  
octets dbch_reserved est un entier sans signe sur 4 octets FIN DEV_BROADCAST_VOLUME est  
structure dbch_size est un entier sans signe sur 4 octets dbch_devicetype est un entier sans  
signe sur 4 octets dbch_reserved est un entier sans signe sur 4 octets dbcv_unitmask est un  
entier sans signe sur 4 octets dbcv_flags est un entier sans signe sur 2 octets FIN nRet est entier  
= 0 // On bloque la popup de notification d&#039;insertion Windows // qui demande ce qu&#039;on  
veut faire du CD/DVD SI wMsg = WM_QUERYCANCELAUTOPLAY ALORS nRet = 1 SINON // On  
gère l&#039;insertion et la suppression d&#039;un device SI wParam =  
DBT_DEVICEREMOVECOMPLETE OU wParam = DBT_DEVICEARRIVAL ALORS SI lParam  
ALORS stDevHDR est DEV_BROADCAST_HDR Transfert(&stDevHDR, lParam,  
Dimension(stDevHDR)) SI stDevHDR:dbch_devicetype = DBT_DEVTYP_VOLUME ALORS  
stDevVOL est DEV_BROADCAST_VOLUME Transfert(&stDevVOL, lParam,
```

```

Dimension(stDevVOL))    Dolt est entier = Faux    sDrive est chaîne =
BFF_DEVICE_DRIVE(stDevVOL:dbcv_unitmask)    hTreeView est entier =
BFF_GET_TV_HANDLE()    sFolderName est chaîne = BFF_GET_PATH()    //
StockageAmovibleRetiré // -----    SI wParam = DBT_DEVICEREMOVECOMPLETE
ALORS    SI Gauche(Majuscule(sFolderName), 1) = Majuscule(sDrive) ALORS    sFolderName =
fRepEnCours()    // Efface le contenu de la table et la liste masquée    ClearFolderContent()
    Dolt = UpdateRegistry(sFolderName)    FIN    // StockageAmovibleInséré // -----
    SINON SI wParam = DBT_DEVICEARRIVAL    SI MsgBoxYesNo("Unité de stockage amovible",
"Une nouvelle unité de stockage a été détectée." + RC + RC + "Souhaitez-vous accéder au média "
+ sDrive + ": ?", UselconDevices) = Oui ALORS    sFolderName = sDrive + ":"    // Efface le
contenu de la table et la liste masquée    ClearFolderContent()    Dolt =
UpdateRegistry(sFolderName)    FIN    FIN    // On masque le Treeview
    ShowWindow(hTreeView, SW_HIDE)    // On réinitialise le TreeView et on se replace sur le
folder actif    InitTreeViewFolder(sFolderName)    // On réaffiche le Treeview
    ShowWindow(hTreeView, SW_SHOW)    // On met à jour la table    SI Dolt ALORS
UpdateTable(MakeLong(0, Bff_SelChange))    FIN    FIN    FIN FIN RENVOYER nRet

```